Analyzing the Lending Club Case Study

Importing the Libraries required for EDA

In [891…

```
#import the required Libararies
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt, seaborn as sns

#adjusting the rows and columns display
pd.set_option('display.max_columns',111)
pd.set_option('display.max_rows',111)
```

1. Reading the Input Data from the File

In [892…

```
#Reading the loan data in pandas
file_path = 'C:/Users/SRSRE/OneDrive - KK-Group/1.Working Files/Desktop/Loan DataSe
loan_df = pd.read_csv(file_path, low_memory = False, parse_dates = ["issue_d"])
loan_df.head()
```

Out[892]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment
0	1077501	1296599	5000	5000	4975.0	36 months	10.65%	162.87
1	1077430	1314167	2500	2500	2500.0	60 months	15.27%	59.83
2	1077175	1313524	2400	2400	2400.0	36 months	15.96%	84.33
3	1076863	1277178	10000	10000	10000.0	36 months	13.49%	339.31
4	1075358	1311748	3000	3000	3000.0	60 months	12.69%	67.79

2. Understanding structure of the Data

In [893…

```
#getting the dataframe dimensions
loan_df.shape
```

Out[893]:

```
(39717, 111)
```

In [894…

```
#getting the column informations
loan_df.dtypes
```

Out[894]:

id	int64
member_id	int64
loan_amnt	int64
funded_amnt	int64
funded_amnt_inv	float64
term	object
int_rate	object
installment	float64
grade	object
sub_grade	object
emp_title	object
emp_length	object
home_ownership	object
annual_inc	int64
verification_status	object
issue_d	datetime64[ns]
loan_status	object
pymnt_plan	object
url	object
desc	object
purpose	object
title	object
zip_code	object
addr_state	object
dti	float64
delinq_2yrs	int64
earliest_cr_line	object
inq_last_6mths	int64
mths_since_last_delinq	float64
mths_since_last_record	float64
open_acc	int64
pub_rec	int64
revol_bal	int64
revol_util	object
total_acc	int64
initial_list_status	object
out_prncp	float64
out_prncp_inv	float64
total_pymnt	float64
total_pymnt_inv	float64
total_rec_prncp	float64
total_rec_int	float64
total_rec_late_fee	float64
recoveries	float64
collection_recovery_fee	float64
last_pymnt_d	object
last_pymnt_amnt	float64
next_pymnt_d	object
last_credit_pull_d	object
collections_12_mths_ex_med	float64
mths_since_last_major_derog	float64
policy_code	int64
application_type	object
annual_inc_joint	float64
dti_joint	float64
verification_status_joint	float64
acc_now_delinq	int64
tot_coll_amt	float64
tot_cur_bal	float64
open_acc_6m	float64
open_il_6m	float64
open_il_12m	float64
open_il_24m	float64
mths_since_rcnt_il	float64

```
total_bal_il                          float64
il_util                               float64
open_rv_12m                           float64
open_rv_24m                           float64
max_bal_bc                            float64
all_util                              float64
total_rev_hi_lim                      float64
inq_fi                                float64
total_cu_tl                           float64
inq_last_12m                          float64
acc_open_past_24mths                  float64
avg_cur_bal                           float64
bc_open_to_buy                        float64
bc_util                               float64
chargeoff_within_12_mths              float64
delinq_amnt                             int64
mo_sin_old_il_acct                    float64
mo_sin_old_rev_tl_op                  float64
mo_sin_rcnt_rev_tl_op                 float64
mo_sin_rcnt_tl                        float64
mort_acc                              float64
mths_since_recent_bc                  float64
mths_since_recent_bc_dlq              float64
mths_since_recent_inq                 float64
mths_since_recent_revol_delinq        float64
num_accts_ever_120_pd                 float64
num_actv_bc_tl                        float64
num_actv_rev_tl                       float64
num_bc_sats                           float64
num_bc_tl                             float64
num_il_tl                             float64
num_op_rev_tl                         float64
num_rev_accts                         float64
num_rev_tl_bal_gt_0                   float64
num_sats                              float64
num_tl_120dpd_2m                      float64
num_tl_30dpd                          float64
num_tl_90g_dpd_24m                    float64
num_tl_op_past_12m                    float64
pct_tl_nvr_dlq                        float64
percent_bc_gt_75                      float64
pub_rec_bankruptcies                  float64
tax_liens                             float64
tot_hi_cred_lim                       float64
total_bal_ex_mort                     float64
total_bc_limit                        float64
total_il_high_credit_limit            float64
dtype: object
```

In [895…

```
#basic info of the data frame
loan_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39717 entries, 0 to 39716
Columns: 111 entries, id to total_il_high_credit_limit
dtypes: datetime64[ns](1), float64(73), int64(14), object(23)
memory usage: 33.6+ MB
```

In [896…

```
#Getting basic statistical details of the data frame
loan_df.describe()
```

Out[896]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	installment	
count	3.971700e+04	3.971700e+04	39717.000000	39717.000000	39717.000000	39717.000000	3
mean	6.831319e+05	8.504636e+05	11219.443815	10947.713196	10397.448868	324.561922	6
std	2.106941e+05	2.656783e+05	7456.670694	7187.238670	7128.450439	208.874874	6
min	5.473400e+04	7.069900e+04	500.000000	500.000000	0.000000	15.690000	4
25%	5.162210e+05	6.667800e+05	5500.000000	5400.000000	5000.000000	167.020000	4
50%	6.656650e+05	8.508120e+05	10000.000000	9600.000000	8975.000000	280.220000	5
75%	8.377550e+05	1.047339e+06	15000.000000	15000.000000	14400.000000	430.780000	8
max	1.077501e+06	1.314167e+06	35000.000000	35000.000000	35000.000000	1305.190000	6

3. Data Quality Check and Missing Values

3.1 Percentage of missing values for columns and rows

In [897…

```
cols = pd.DataFrame(loan_df.isnull().mean().round(4) * 100, columns = ['percentage_
print(cols)
```

	percentage_missing_value
id	0.00
earliest_cr_line	0.00
inq_last_6mths	0.00
open_acc	0.00
pub_rec	0.00
revol_bal	0.00
total_acc	0.00
initial_list_status	0.00
out_prncp	0.00
out_prncp_inv	0.00
delinq_2yrs	0.00
total_pymnt	0.00
total_rec_int	0.00
total_rec_late_fee	0.00
recoveries	0.00
collection_recovery_fee	0.00
last_pymnt_amnt	0.00
policy_code	0.00
application_type	0.00
acc_now_delinq	0.00
delinq_amnt	0.00
total_pymnt_inv	0.00
dti	0.00
total_rec_prncp	0.00
zip_code	0.00
member_id	0.00
loan_amnt	0.00
addr_state	0.00
funded_amnt_inv	0.00
term	0.00
int_rate	0.00
installment	0.00
grade	0.00
sub_grade	0.00
home_ownership	0.00
annual_inc	0.00
funded_amnt	0.00
issue_d	0.00
purpose	0.00
verification_status	0.00
loan_status	0.00
pymnt_plan	0.00
url	0.00
last_credit_pull_d	0.01
title	0.03
tax_liens	0.10
revol_util	0.13
collections_12_mths_ex_med	0.14
chargeoff_within_12_mths	0.14
last_pymnt_d	0.18
pub_rec_bankruptcies	1.75
emp_length	2.71
emp_title	6.19
desc	32.58
mths_since_last_delinq	64.66
mths_since_last_record	92.99
next_pymnt_d	97.13
num_bc_sats	100.00
mths_since_recent_bc	100.00
mths_since_recent_bc_dlq	100.00
mths_since_recent_inq	100.00
mths_since_recent_revol_delinq	100.00
num_accts_ever_120_pd	100.00

```
num_actv_bc_tl                                100.00
num_actv_rev_tl                               100.00
mort_acc                                      100.00
num_bc_tl                                     100.00
num_tl_op_past_12m                            100.00
num_op_rev_tl                                 100.00
num_rev_accts                                 100.00
num_rev_tl_bal_gt_0                           100.00
num_sats                                      100.00
num_tl_120dpd_2m                              100.00
num_tl_30dpd                                  100.00
num_tl_90g_dpd_24m                            100.00
pct_tl_nvr_dlq                                100.00
percent_bc_gt_75                              100.00
tot_hi_cred_lim                               100.00
total_bal_ex_mort                             100.00
mo_sin_rcnt_tl                                100.00
num_il_tl                                     100.00
mo_sin_rcnt_rev_tl_op                         100.00
verification_status_joint                     100.00
mo_sin_old_il_acct                            100.00
mths_since_last_major_derog                   100.00
annual_inc_joint                              100.00
dti_joint                                     100.00
total_bc_limit                                100.00
tot_coll_amt                                  100.00
tot_cur_bal                                   100.00
open_acc_6m                                   100.00
open_il_6m                                    100.00
open_il_12m                                   100.00
open_il_24m                                   100.00
mths_since_rcnt_il                            100.00
total_bal_il                                  100.00
il_util                                       100.00
open_rv_12m                                   100.00
open_rv_24m                                   100.00
max_bal_bc                                    100.00
all_util                                      100.00
total_rev_hi_lim                              100.00
inq_fi                                        100.00
total_cu_tl                                   100.00
inq_last_12m                                  100.00
acc_open_past_24mths                          100.00
avg_cur_bal                                   100.00
bc_open_to_buy                                100.00
bc_util                                       100.00
mo_sin_old_rev_tl_op                          100.00
total_il_high_credit_limit                    100.00
```

In [898…

```
#summary of missing values associated with columns
print(str(round(100.0 * cols[cols['percentage_missing_value']==0].count()/len(cols)
print(str(round(100.0 * cols[(cols['percentage_missing_value']>0) & (cols['percenta
print(str(round(100.0 * cols[(cols['percentage_missing_value']>10) & (cols['percent
print(str(round(100.0 * cols[cols['percentage_missing_value']>50].count()/len(cols)
```

```
percentage_missing_value    38.74
dtype: float64% columns have no missing value
percentage_missing_value    9.01
dtype: float64% columns have missing value betwee 0-10%
percentage_missing_value    0.9
dtype: float64% columns have missing value betwee 10-50%
percentage_missing_value    51.35
dtype: float64% columns have more than 50% missing value
```

In [899…
```
#checking row-wise null percentages
row_null = pd.DataFrame(loan_df.isnull().sum(axis =1),columns = ['num_missing_value
row_null
```

Out[899]:

	num_missing_value
0	58
1	57
2	59
3	56
4	55
...	...
39712	59
39713	59
39714	61
39715	61
39716	59

39717 rows × 1 columns

3.2 Removing the columns with high percentage of missing values (>50%)

In [900…
```
#removing columns where we have >50% of the values are null
threshold = 0.5
percentage_null_values = (loan_df.isnull().mean() * 100).round(2)
columns_to_remove = percentage_null_values[percentage_null_values > threshold].inde
loan_df = loan_df.drop(columns=columns_to_remove)
```

In [901…
```
#checking the datafrme again afer removing the columns where >50% the values are nu
null_values_per_column = loan_df.isnull().sum()
print(null_values_per_column)
```

```
id                            0
member_id                     0
loan_amnt                     0
funded_amnt                   0
funded_amnt_inv               0
term                          0
int_rate                      0
installment                   0
grade                         0
sub_grade                     0
home_ownership                0
annual_inc                    0
verification_status           0
issue_d                       0
loan_status                   0
pymnt_plan                    0
url                           0
purpose                       0
title                        11
zip_code                      0
addr_state                    0
dti                           0
delinq_2yrs                   0
earliest_cr_line              0
inq_last_6mths                0
open_acc                      0
pub_rec                       0
revol_bal                     0
revol_util                   50
total_acc                     0
initial_list_status           0
out_prncp                     0
out_prncp_inv                 0
total_pymnt                   0
total_pymnt_inv               0
total_rec_prncp               0
total_rec_int                 0
total_rec_late_fee            0
recoveries                    0
collection_recovery_fee       0
last_pymnt_d                 71
last_pymnt_amnt               0
last_credit_pull_d            2
collections_12_mths_ex_med    56
policy_code                   0
application_type              0
acc_now_delinq                0
chargeoff_within_12_mths      56
delinq_amnt                   0
tax_liens                     39
dtype: int64
```

In [902…

```
#getting the dataframe dimensions after removing columns with >90% values are null
loan_df.shape
```

Out[902]:

```
(39717, 50)
```

In [903…

```
# re-checking columns with missing
round(100.0 * loan_df.isnull().sum()/len(loan_df),2).sort_values()
```

Out[903]:
```
id                          0.00
delinq_amnt                 0.00
open_acc                    0.00
pub_rec                     0.00
revol_bal                   0.00
total_acc                   0.00
initial_list_status         0.00
out_prncp                   0.00
out_prncp_inv               0.00
total_pymnt                 0.00
total_pymnt_inv             0.00
total_rec_prncp             0.00
total_rec_int               0.00
total_rec_late_fee          0.00
recoveries                  0.00
collection_recovery_fee     0.00
last_pymnt_amnt             0.00
policy_code                 0.00
application_type            0.00
acc_now_delinq              0.00
earliest_cr_line            0.00
delinq_2yrs                 0.00
inq_last_6mths              0.00
addr_state                  0.00
member_id                   0.00
loan_amnt                   0.00
funded_amnt                 0.00
funded_amnt_inv             0.00
term                        0.00
int_rate                    0.00
installment                 0.00
dti                         0.00
sub_grade                   0.00
home_ownership              0.00
grade                       0.00
verification_status         0.00
issue_d                     0.00
loan_status                 0.00
pymnt_plan                  0.00
url                         0.00
purpose                     0.00
zip_code                    0.00
annual_inc                  0.00
last_credit_pull_d          0.01
title                       0.03
tax_liens                   0.10
revol_util                  0.13
collections_12_mths_ex_med  0.14
chargeoff_within_12_mths    0.14
last_pymnt_d                0.18
dtype: float64
```

3.3 Subsetting the data to filter only the defaulters data

In [904…
```
#creatting subset of the data for only defualted customers for further steps
loan_df_Chargedoff = loan_df[loan_df['loan_status'] == 'Charged Off']
```

In [905…
```
#getting the dataframe dimensions after subsetting the data to only the defaulters
loan_df_Chargedoff.shape
```

Out[905]:
```
(5627, 50)
```

In [906…

```
#getting the column informations
loan_df_Chargedoff.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5627 entries, 1 to 39688
Data columns (total 50 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   id                          5627 non-null   int64
 1   member_id                   5627 non-null   int64
 2   loan_amnt                   5627 non-null   int64
 3   funded_amnt                 5627 non-null   int64
 4   funded_amnt_inv             5627 non-null   float64
 5   term                        5627 non-null   object
 6   int_rate                    5627 non-null   object
 7   installment                 5627 non-null   float64
 8   grade                       5627 non-null   object
 9   sub_grade                   5627 non-null   object
 10  home_ownership              5627 non-null   object
 11  annual_inc                  5627 non-null   int64
 12  verification_status         5627 non-null   object
 13  issue_d                     5627 non-null   datetime64[ns]
 14  loan_status                 5627 non-null   object
 15  pymnt_plan                  5627 non-null   object
 16  url                         5627 non-null   object
 17  purpose                     5627 non-null   object
 18  title                       5625 non-null   object
 19  zip_code                    5627 non-null   object
 20  addr_state                  5627 non-null   object
 21  dti                         5627 non-null   float64
 22  delinq_2yrs                 5627 non-null   int64
 23  earliest_cr_line            5627 non-null   object
 24  inq_last_6mths              5627 non-null   int64
 25  open_acc                    5627 non-null   int64
 26  pub_rec                     5627 non-null   int64
 27  revol_bal                   5627 non-null   int64
 28  revol_util                  5611 non-null   object
 29  total_acc                   5627 non-null   int64
 30  initial_list_status         5627 non-null   object
 31  out_prncp                   5627 non-null   float64
 32  out_prncp_inv               5627 non-null   float64
 33  total_pymnt                 5627 non-null   float64
 34  total_pymnt_inv             5627 non-null   float64
 35  total_rec_prncp             5627 non-null   float64
 36  total_rec_int               5627 non-null   float64
 37  total_rec_late_fee          5627 non-null   float64
 38  recoveries                  5627 non-null   float64
 39  collection_recovery_fee     5627 non-null   float64
 40  last_pymnt_d                5556 non-null   object
 41  last_pymnt_amnt             5627 non-null   float64
 42  last_credit_pull_d          5626 non-null   object
 43  collections_12_mths_ex_med  5621 non-null   float64
 44  policy_code                 5627 non-null   int64
 45  application_type            5627 non-null   object
 46  acc_now_delinq              5627 non-null   int64
 47  chargeoff_within_12_mths    5621 non-null   float64
 48  delinq_amnt                 5627 non-null   int64
 49  tax_liens                   5626 non-null   float64
dtypes: datetime64[ns](1), float64(16), int64(14), object(19)
memory usage: 2.2+ MB
```

In [907…

```
#Creating the list of Categorical Columns
Categorical_columns = ['term','grade','sub_grade','home_ownership',
                       'purpose','addr_state']
```

In [908…

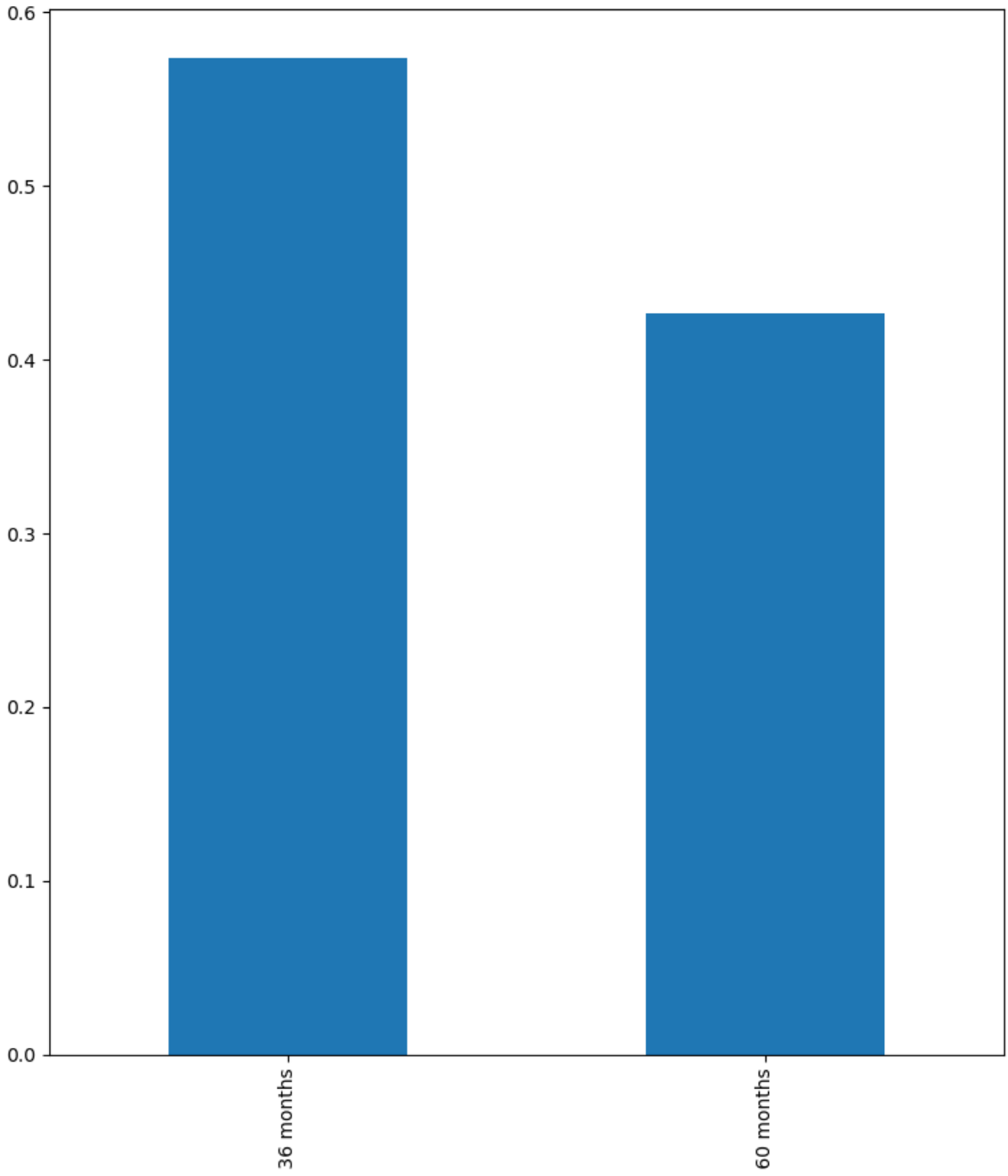
```
#Creating the list of numerical columns
Numerical_columns = ['loan_amnt','funded_amnt','funded_amnt_inv','installment','ann
                     'open_acc','pub_rec','revol_bal','total_acc','total_pymnt','tot
                     ,'recoveries','collection_recovery_fee','last_pymnt_amnt']
```

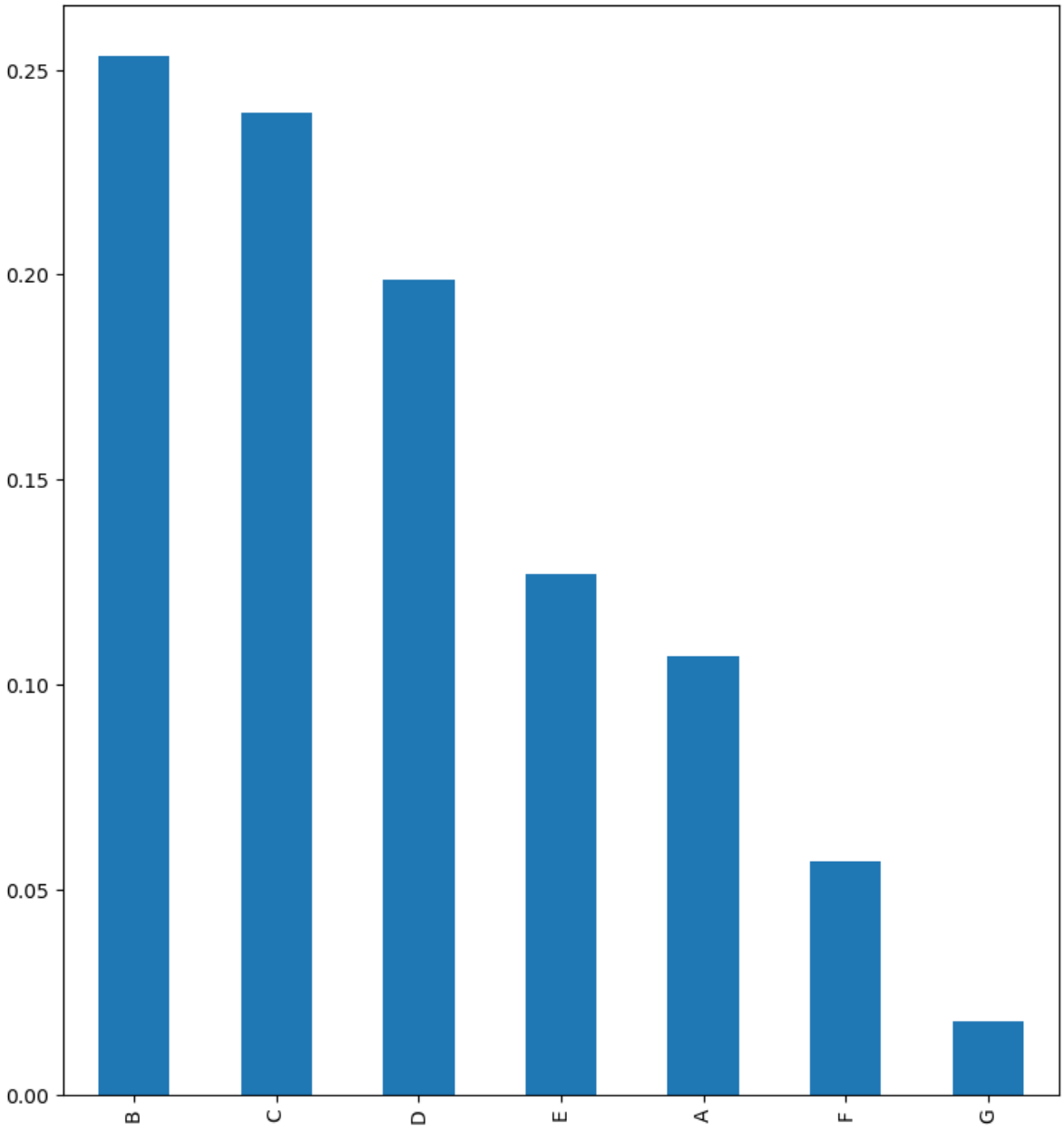
4 Univariate Analysis

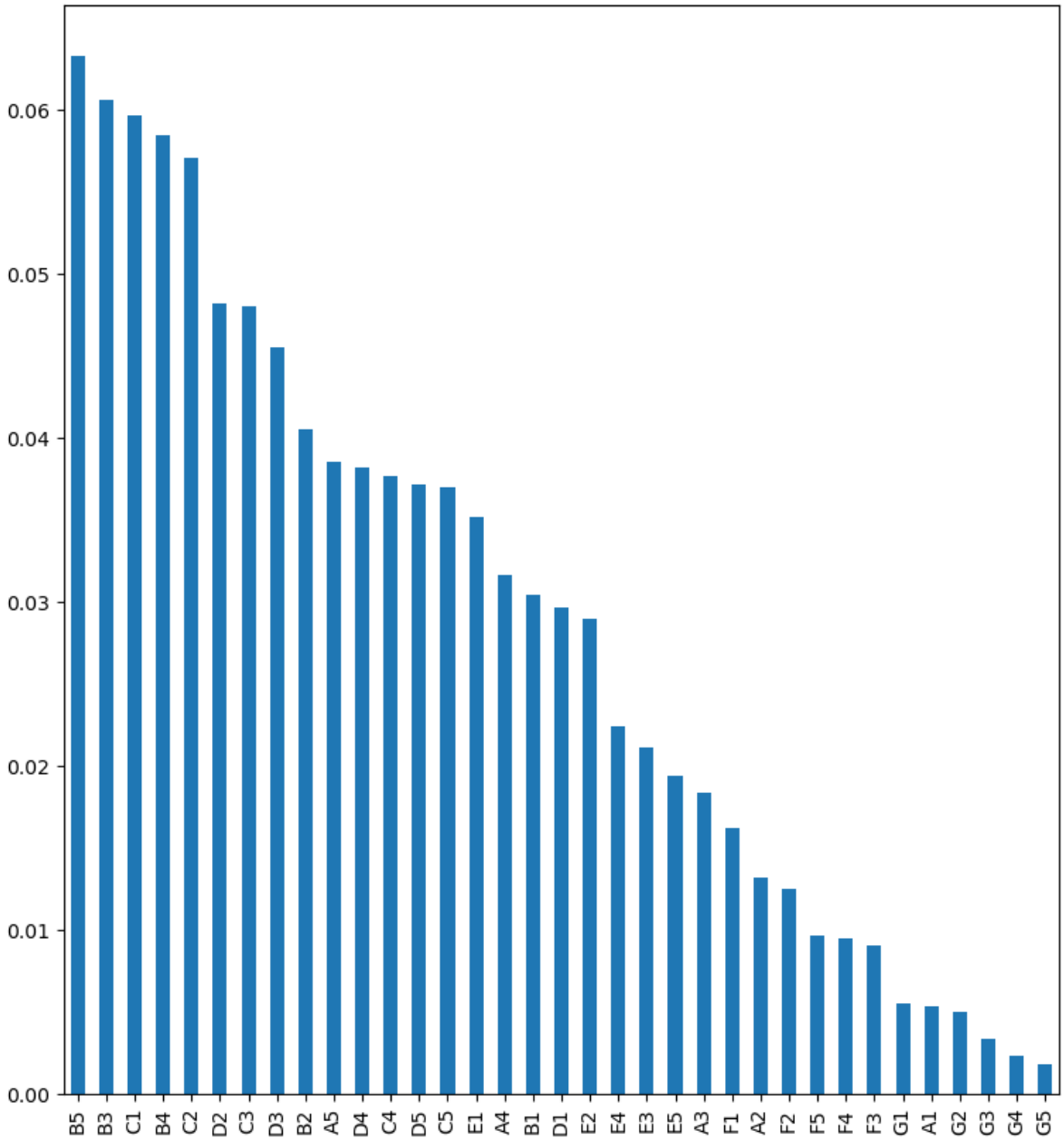
Under univariate analysis, we will look at the percentage of distribution of values of categorical variable

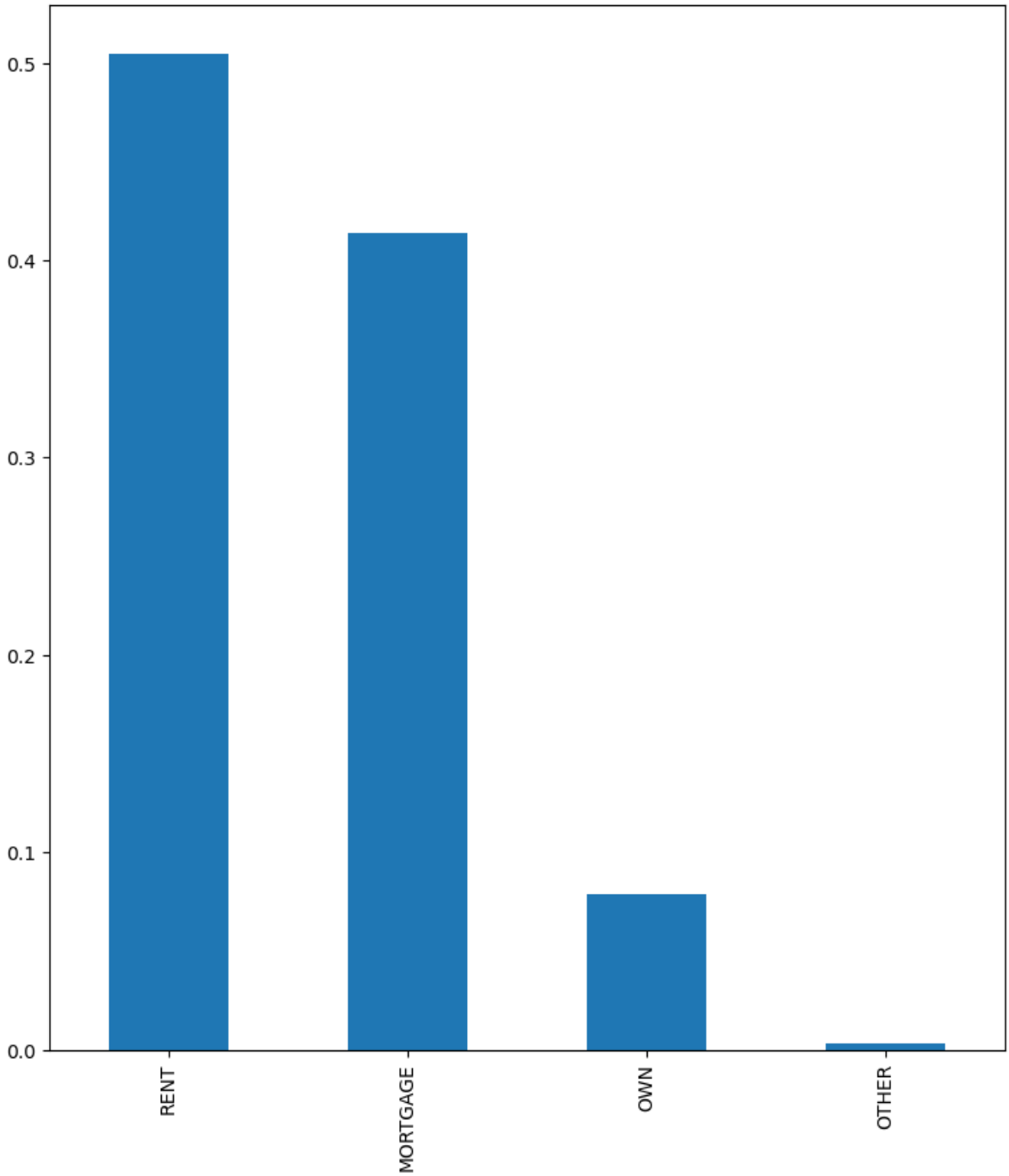
In [909…

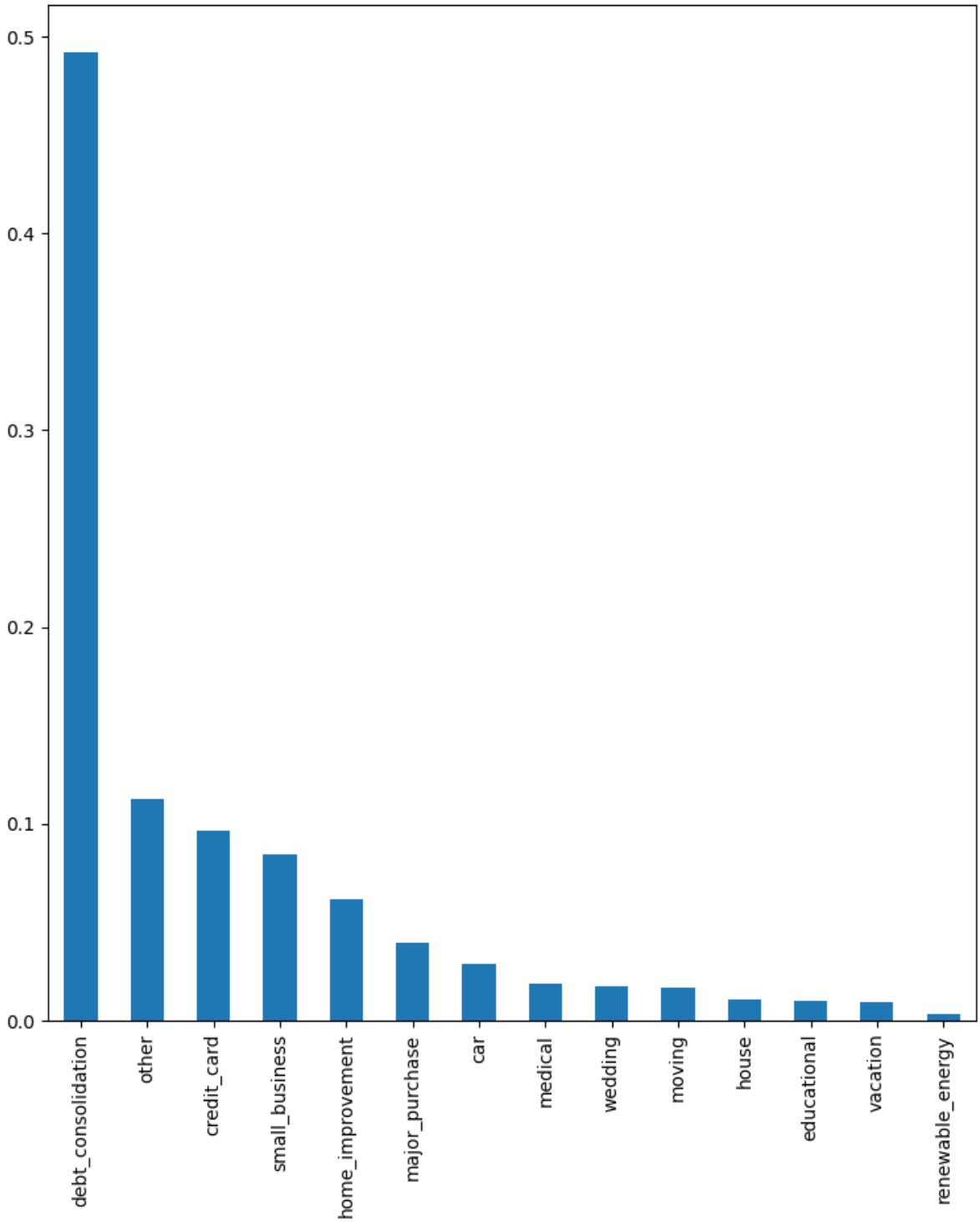
```
for i in Categorical_columns :
    plt.figure(figsize = (20,10))
    plt.subplot(1,2,1)
    loan_df_Chargedoff[i].value_counts(normalize = True).plot.bar()
```

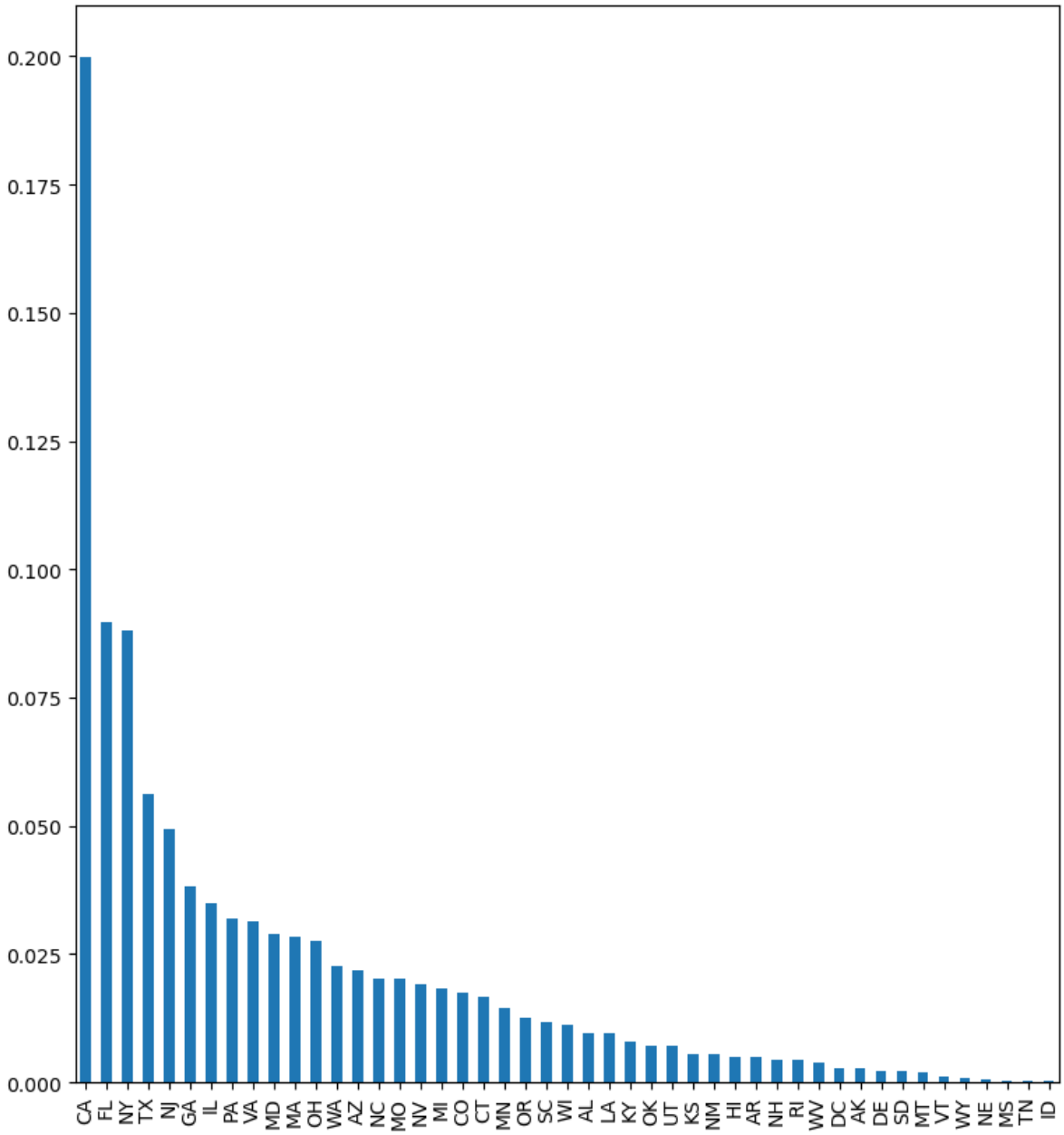



0.25
0.20
0.15
0.10
0.05
0.00
B
C
D
E
A
F
G


0.06
0.05
0.04
0.03
0.02
0.01
0.00
B5
B3
C1
B4
C2
D2
C3
D3
B2
A5
D4
C4
D5
C5
E1
A4
B1
D1
E2
E4
E3
E5
A3
F1
A2
F2
F5
F4
F3
G1
A1
G2
G3
G4
G5

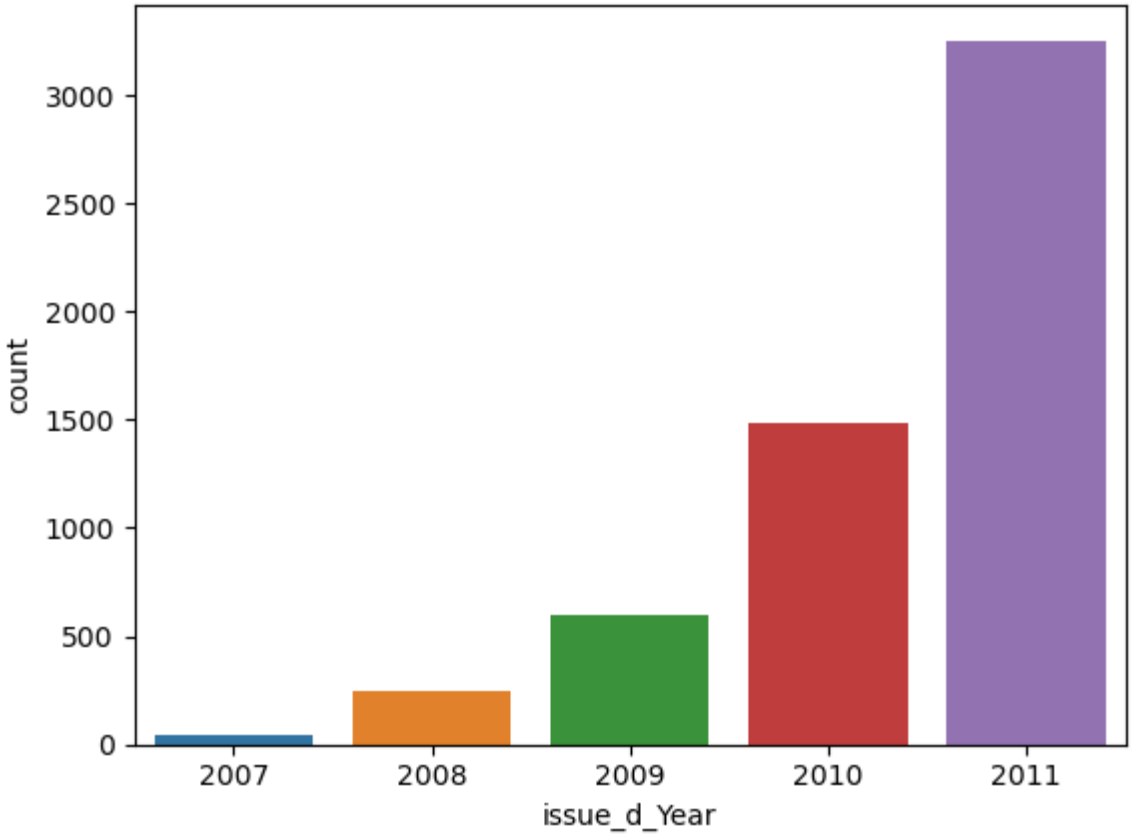

0.5
0.4
0.3
0.2
0.1
0.0
RENT
MORTGAGE
OWN
OTHER


0.5
0.4
0.3
0.2
0.1
0.0
debt_consolidation
other
credit_card
small_business
home_improvement
major_purchase
car
medical
wedding
moving
house
educational
vacation
renewable_energy



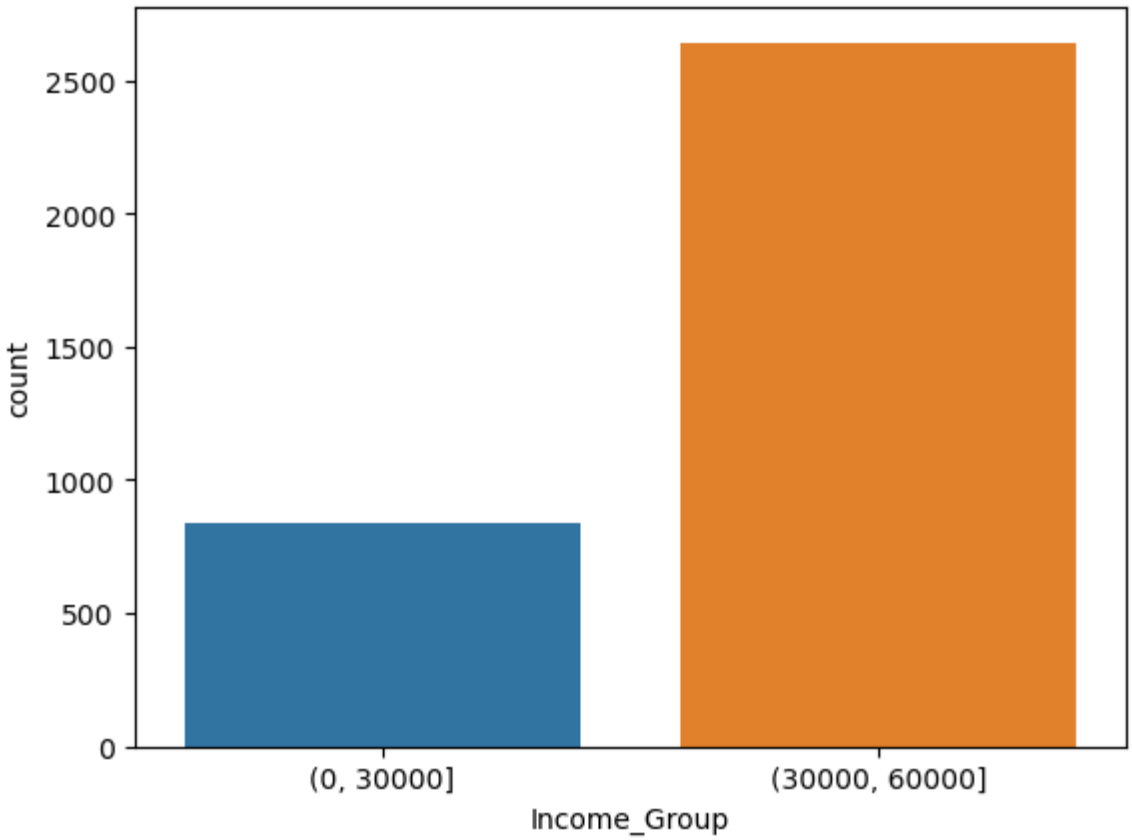
In [910…

```
#converting issue_d to a date datatype
loan_df_Chargedoff["issue_d"] = pd.to_datetime(loan_df_Chargedoff["issue_d"])

#creating a new year derived column to see year on year defaulters trend
loan_df_Chargedoff["issue_d_Year"]= loan_df_Chargedoff["issue_d"] .dt.year

#creating a plot to see the Year on Year trend on the defaulters
sns.countplot(data = loan_df_Chargedoff, x ="issue_d_Year")
plt.show()
```



In [911…

```
#creating a derived column to grouop the income groups
bins = [0,30000,60000]
lables = ['<30000','>30000',float('inf')]
loan_df_Chargedoff["Income_Group"] = pd.cut(loan_df_Chargedoff["annual_inc"], bins=

#creating a countplot to visulize the income group defaulters
sns.countplot(data = loan_df_Chargedoff, x ="Income_Group")
plt.show()
```



Key Interpretation from univariate analysis of categorical variables

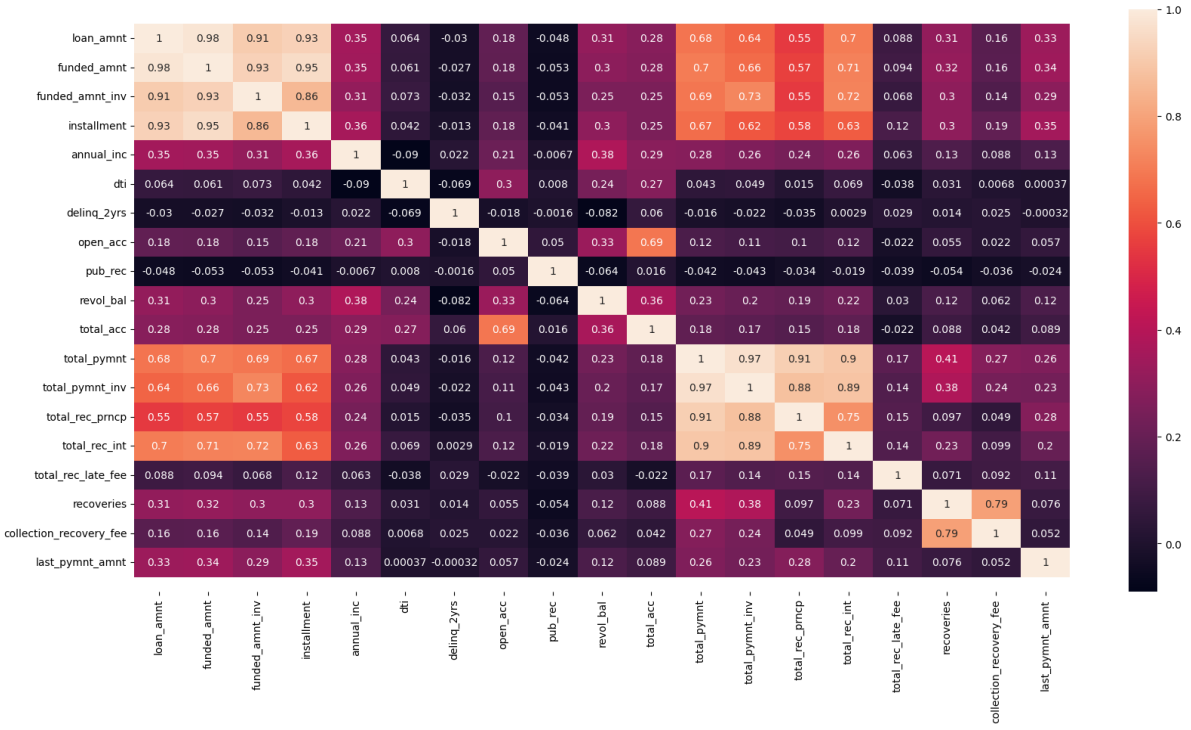
Important observations

1. term : Loans with a 36-month term have a higher default rate compared to those with a 60-month term.
2. Grade : Loans with grades B, C, and D have a higher default rate compared to loans with other grades.
3. sub_grade : The top 5 subgrades with the highest default rates are B5, B3, C1, B4, and C2.
4. home_ownership : Borrowers who own their homes have a lower default rate compared to those who do not own a home.
5. purpose : Loans taken for the purpose of debt consolidation have the highest default rate.
6. state : The state CA has the highest number of defaulters among all states.
7. issue_d : The number of defaulters has been steadily increasing from year to year.
8. income : Individuals earning more than 30,000 are more inclined to default on their loans compared to those earning less than 30,000

Correlation for numerical columns

In [912…

```
plt.figure(figsize = (20,10))
sns.heatmap(loan_df_Chargedoff[Numerical_columns].corr(),annot = True)
b, t = plt.ylim()
b += 0.5
t -= 0.5
plt.ylim(b, t)
plt.yticks(rotation = 0)
plt.show()
```



Important observations

1. loan_amnt, funded_amnt, funded_amnt_inv& installment columns are highly correalted
2. total_pymmnt, total_payment_inv,total_rec_prncp,total_rec_int are moderatly correlated to point 1 columns
3. for the further analysis we could reduce the features which are highly correlated

In [ ]: